**实验五 类与对象**

**【实验目的与要求】**

1、掌握声明类的方法，类和类的成员的概念以及定义对象的方法。

2、初步掌握用类和对象编制基于对象的程序。

3、学习检查和调试基于对象的程序。

**【实验内容】**

1、有以下程序：

#include<iostream>

using namespace std；

class Time // 定义Time类

{

public： // 数据成员为公用的

int hour;

int minute;

int sec ;

};

int main()

{

Time tl; //定义t1为Time类对象

cin>>t1.hour; //输入设定的时间

cin>>t1.minute;

cin>>t1.sec;

cout<<t1.hourl<<”：”<<t1.minute<<”：”<<t1.sec<<endl ;

return 0;

}

改写程序，要求：

(1)将数据成员改为私有的；

1. 将输入和输出的功能改为由成员函数实现；

(3)在类体内定义成员函数。

然后编译和运行程序。请分析什么成员应指定为公用的?什么成员应指定为私有的? 什么函数最好放在类中定义? 什么函数最好在类外定义?

1.用户使用类的公用接口，如公用数据成员和公用成员函数。

2.类的内部实现细节和一些隐私内容，包括私有数据成员和私有成员函数。

3.与类的数据成员直接相关的函数最好放在类中定义，以便访问和操作类的数据成员。

4.与类的数据成员无关的函数，或者与类的其他成员函数无关的函数，最好在类外定义。

#include <iostream>

using namespace std;

class Time // 定义Time类

{

public:

void Settime()

{

cout << "输入时间" << endl;

cin >> this->hour;

cin >> this->minute;

cin >>this->sec;

}

//将输入和输出的功能改为由成员函数实现

//在类体内定义成员函数

void Printtime()

{

cout << "时间为：";

cout << this->hour << "：" << this->minute << "：" << this->sec << endl;

}

private:// 数据成员为私有的

int hour;

int minute;

int sec;

};

int main()

{

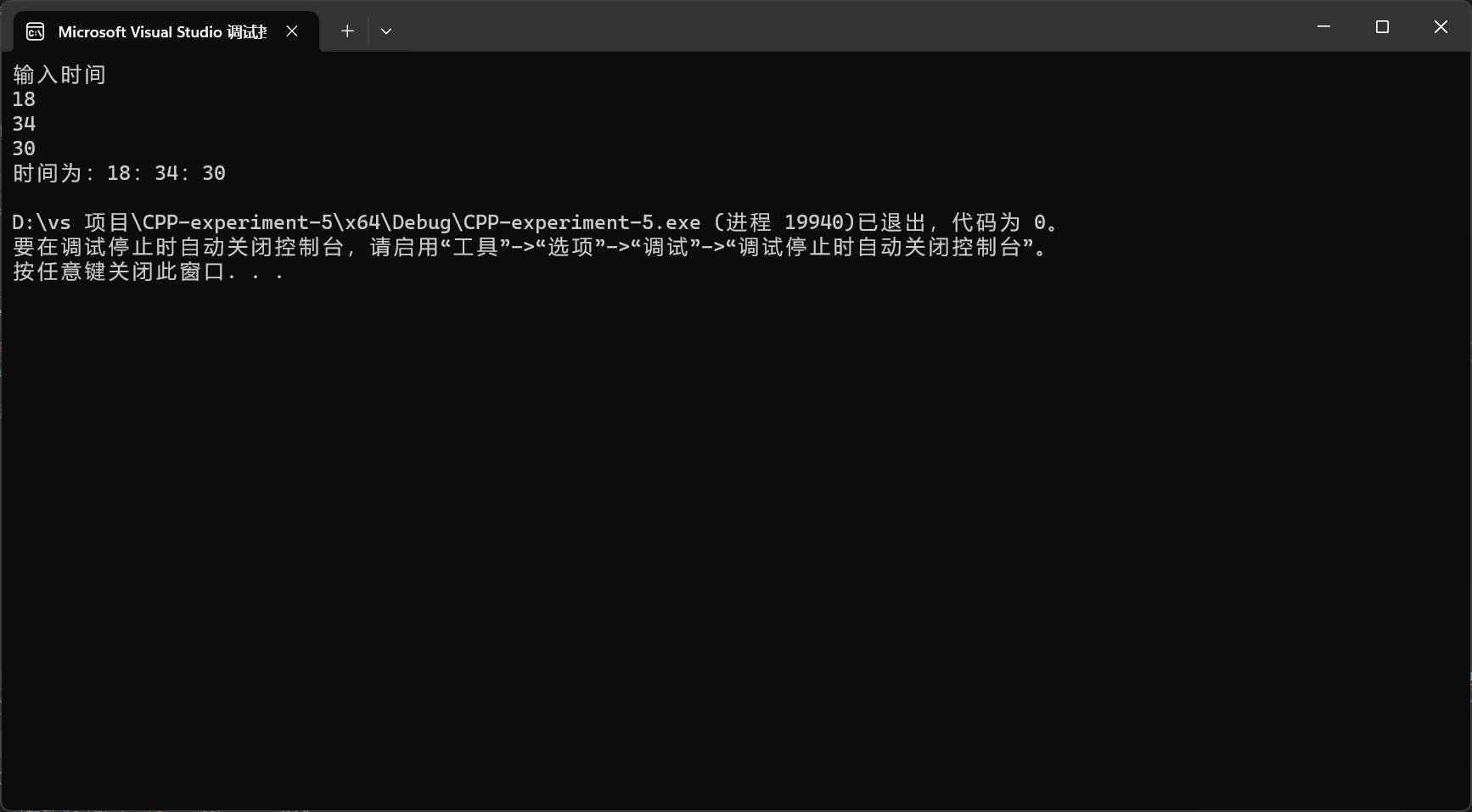
Time t1; //定义t1为Time类对象

t1.Settime();

t1.Printtime();

return 0;

}



2、分别给出如下的3个文件：

(1)含类定义的头文件student.h，

//student.h (这是头文件，在此文件中进行类的声明)

class Student //类声明

{

public: //公用成员函数原型声明

void display();

private:

int num;

char name[20];

char sex;

};

(2)包含成员函数定义的源文件student.cpp

//student.cpp 在此文件中进行函数的定义

#include <iostream>

#include”student.h” //不要漏写此行，否则编译通不过

void Student::display() //在类外定义display类函数

{

cout<<”num：”<<num<<endl;

cout<<”name：”<<name<<endl;

cout<<”sex：”<<sex<<endl;

}

(3)包含主函数的源文件main.cpp。

为了组成一个完整的源程序，应当有包括主函数的源文件：

//main.cpp 主函数模块

#include <iostream> //将类声明头文件包含进来

#include “student.h”

int main()

{

Student stud; //定义对象

Student stud1(007,”tcg”,’m’);

stud.display(); //执行stud对象的display函数

return 0;

}

请完善该程序，在类中增加一个对数据成员赋初值的成员函数set\_value。上机调试并运行。

#pragma once

class Student //类声明

{

public: //公用成员函数原型声明

void display();

void set\_value(int num,const char\* name,char sex);

private:

int num;

char name[20];

char sex;

};

#include <iostream>

#include "student.h" //不要漏写此行，否则编译通不过

using namespace std;

void Student::display() //在类外定义display类函数

{

cout << "num：" << num << endl;

cout << "name：" << name << endl;

cout << "sex：" << sex << endl;

}

void Student::set\_value(int num,const char\* name,char sex)

{

this->num = num;

strcpy\_s(this->name,name);

this->sex = sex;

}

#include <iostream> //将类声明头文件包含进来

#include "student.h"

int main()

{

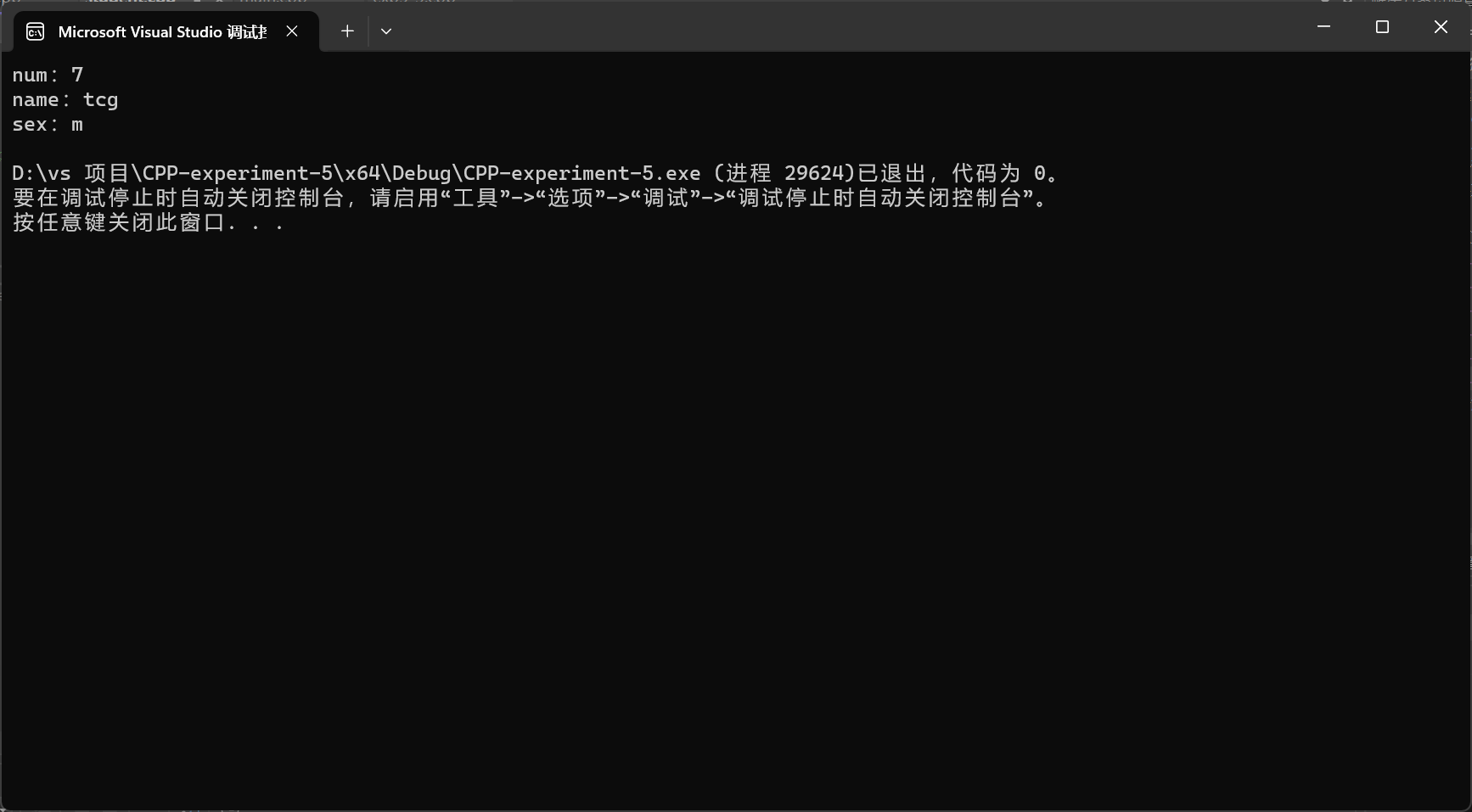
Student stud; //定义对象

stud.set\_value(007, "tcg", 'm');

stud.display(); //执行stud对象的display函数

return 0;

}



3、需要求3个长方柱的体积，请编一个基于对象的程序。数据成员包括length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

(1)由键盘分别输入3个长方柱的长、宽、高；

(2)计算长方柱的体积；

(3)输出3个长方柱的体积。

请编程序，上机调试并运行。

#include <iostream>

using namespace std;

class RTG

{

public:

void Sert()

{

cout << "请依次输入长方柱的长、宽和高" << endl;

cin >> length;

cin >> width;

cin >> height;

}

void PrintVec()

{

cout << "长方柱体积为：";

cout << length \* width \* height << endl;

}

private:

int length;

int width;

int height;

};

int main()

{

RTG r1;

RTG r2;

RTG r3;

r1.Sert();

r2.Sert();

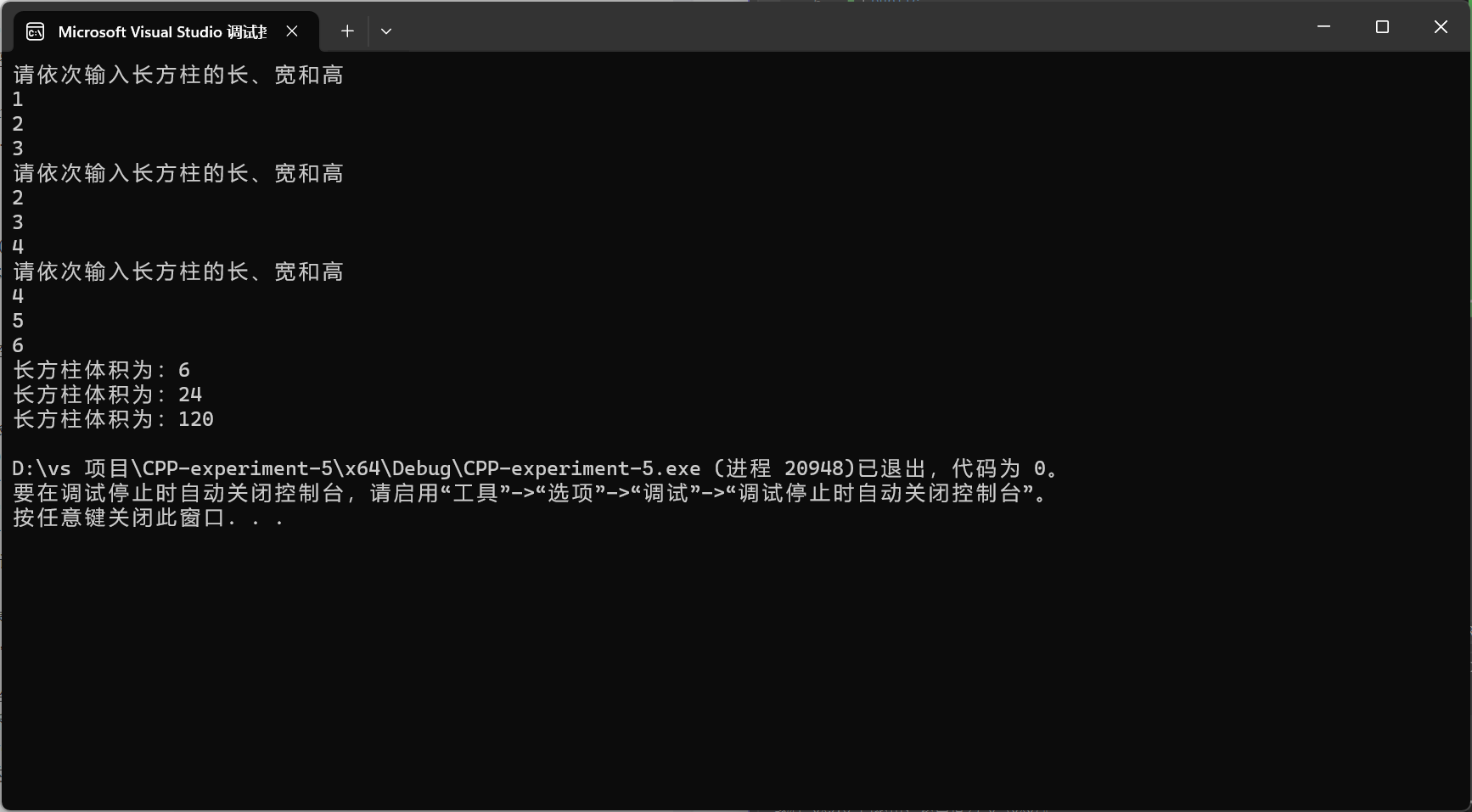
r3.Sert();

r1.PrintVec();

r2.PrintVec();

r3.PrintVec();

}



4、建立一个对象数组，内放5个学生的（学号，成绩），设立一个函数max，用指向对象的指针作函数参数，在max函数中找出5个学生的最高成绩者，并输出其学号。

#include <iostream>

using namespace std;

class Student

{

public:

Student(int Sid, int Score)

{

this->Sid = Sid;

this->Score = Score;

}

int Sid;

int Score;

};

void max(Student\* s)

{

int maxSid = s[0].Sid;

int maxScore = s[0].Score;

for (int i = 0; i < 4; i++)

{

if (s[i].Score < s[i + 1].Score)

{

maxSid = s[i + 1].Sid;

maxScore = s[i + 1].Score;

}

}

cout << "成绩最高者的成绩为：" << maxScore << endl;

cout << "成绩最高者的学号为：" << maxSid << endl;

}

int main()

{

Student student[5] =

{

Student(1,100),

Student(2,99),

Student(3,98),

Student(4,97),

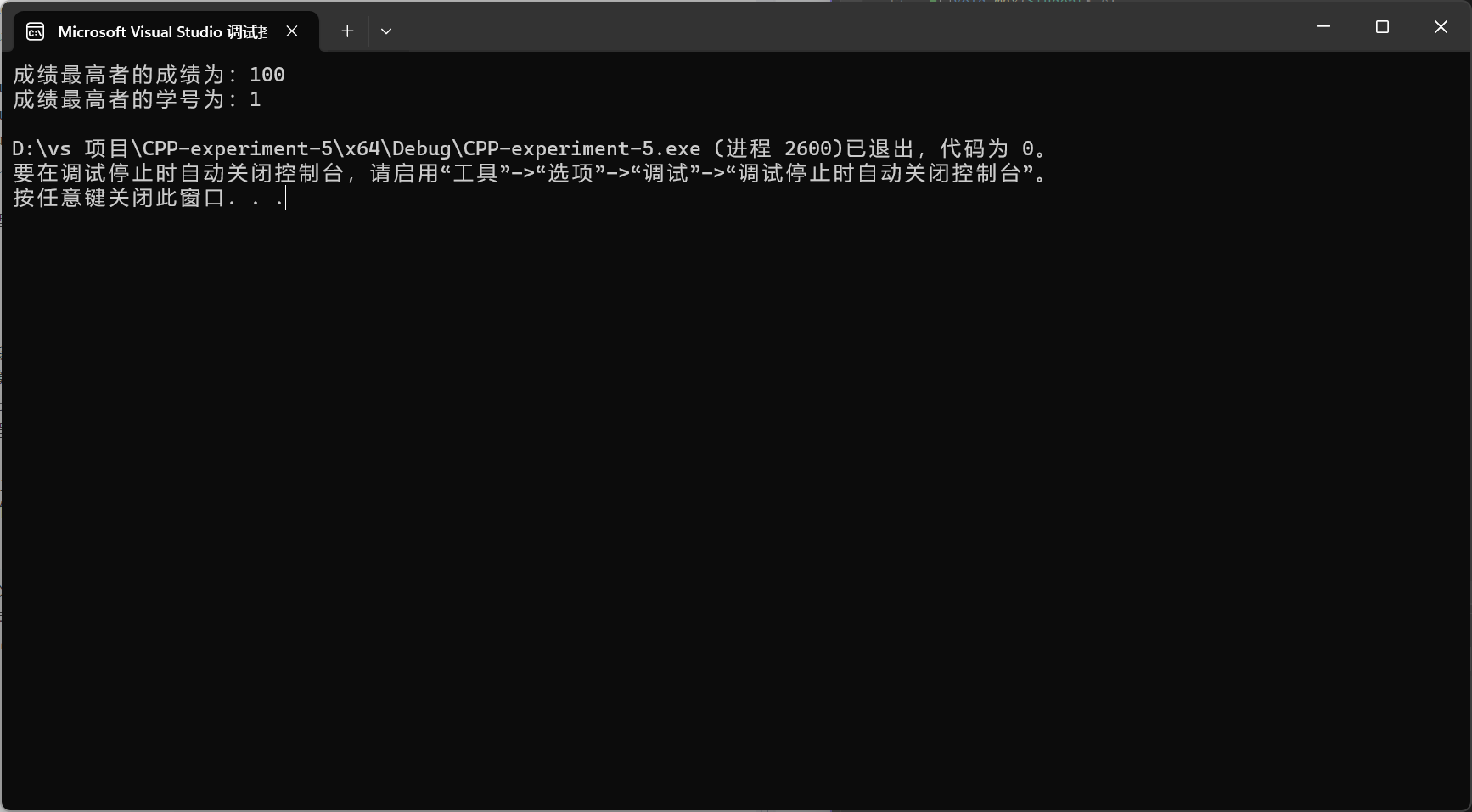
Student(5,96),

};

max(student);

return 0;

}



1. 设有一描述坐标点的类Point，其私有变量x和y代表一个点的(x,y)坐标值。请编写程序实现以下功能：利用构造函数传递参数，在定义对象时将x、y坐标值初始化为（60,80）；利用公有成员函数void setPoint(int i, int j)将坐标值修改为(60+i,80+j)；利用公有成员函数display()输出修改后的坐标值。主函数中通过定义对象，验证各个函数。

#include <iostream>

using namespace std;

class Point

{

public:

void setPoint(int i, int j)

{

this->x = x +i;

this->y = y + j;

}

void display()

{

cout << "修改后的坐标值为" << "(" << x << "," << y << ")" << endl;

}

private:

int x = 60;

int y = 80;

};

int main()

{

Point p;

p.setPoint(10,20);

p.display();

return 0;

}

